

Python: module genutil.statistics

genutil.statistics

[index](#)

Modules

[MA](#)
[MV](#)

[Numeric](#)
[cdms](#)

[cdutil](#)
[genutil](#)

Classes

[exceptions.Exception](#)
[StatisticsError](#)

class **StatisticsError**([exceptions.Exception](#))

Methods defined here:

`__init__`(self, args=None)
Create an exception

`__repr__` = **`__str__`**(self)

`__str__`(self)
Calculate the string representation

Methods inherited from [exceptions.Exception](#):

`__getitem__`(...)

Functions

`autocorrelation`(x, lag=None, axis=0, centered=1, partial=1, biased=1, noloop=0, max_pct_missing=100.0)

Function: autocorrelation

Description of function:

Returns the autocorrelation of a slab at lag k centered, partial
"biased" by default

Usage:

result = [autocorrelation](#)(x, lag=lagoptions, axis=axisoptions,

centered=centeredoptions, partial=partialoptions,
biased=biasedoptions, noloop=noloopoptions)

Options:

lagoptions None | n | (n1, n2, n3...) | [n1, n2, n3]
default value = None the maximum possible lags for specification
is used. You can pass an integer, list of integers, or tuple of integers.
axisoptions 'x' | 'y' | 'z' | 't' | '(dimension_name)' | 0 | 1
default value = 0. You can pass the name of the dimension (integer value 0...n) over which you want to compute the statistic.
centeredoptions None | 0 | 1
default value = 1 removes the mean first. Set to 0 or None to compute
uncentered statistic.
partialoptions None | 0 | 1
default value = 1 uses only common time for means.
biasedoptions None | 0 | 1
default value = 1 computes the biased statistic. If want to compute
an unbiased statistic pass anything but 1.
noloopoptions None | 0 | 1
default value = 0 computes statistic at all lags upto 'lag'.
set noloop=1 statistic is computed at lag only (not up to 'lag').
max_pct_missingoptions
default value = 100. Maximum fraction of cell which is all masked.
Set to a percentage between 0 and 100%.
Set to 0. to mask results if any data is masked.
Set to 100. to calculate result if any data is not masked.

autocovariance(x, lag=None, axis=0, centered=1, partial=1, noloop=0, max_pct_missing=100.0)

Function: autocovariance

Description of function:

Returns the autocovariance of a slab. By default over the first dimension.

Usage:

```
result = autocovariance(x, lag=lagoptions, axis=axisoptions,  
                        centered=centeredoptions,  
                        partial=partialoptions, noloop=noloopoptions)
```

Options:

lagoptions None | n | (n1, n2, n3...) | [n1, n2, n3]
default value = None the maximum possible lags for specification
is used. You can pass an integer, list of integers, or tuple of integers.
axisoptions 'x' | 'y' | 'z' | 't' | '(dimension_name)' | 0 | 1
default value = 0. You can pass the name of the dimension (integer value 0...n) over which you want to compute the statistic.
centeredoptions None | 0 | 1
default value = 1 removes the mean first. Set to 0 or None to compute
uncentered statistic.
partialoptions None | 0 | 1
default value = 1 uses only common time for means.
noloopoptions None | 0 | 1
default value = 0 computes statistic at all lags upto 'lag'.
set noloop=1 statistic is computed at lag only (not up to 'lag').
max_pct_missingoptions
default value = 100. Maximum fraction of cell which is all masked.

Set to a percentage between 0 and 100%.
Set to 0. to mask results if any data is masked.
Set to 100. to calculate result if any data is not masked

checker(x, weights=None, axis=0, centered=1)

correlation(x, y, weights=None, axis=0, centered=1, biased=1, max_pct_missing=100.0)

Function: correlation

Description of function:

Returns the correlation between 2 slabs. By default on the first dimension, centered and biased by default.

Usage:

```
result = correlation(x, y, weights=weightoptions, axis=axisoptions,
                      centered=centeredoptions, biased=biasedoptions,
                      max_pct_missing=max_pct_missingoptions)
```

Options:

weightoptions
default = None. If you want to compute the weighted correlation, provide the weights here.

axisoptions 'x' | 'y' | 'z' | 't' | '(dimension_name)' | 0 | 1
default value = 0. You can pass the name of the dimension (integer value 0...n) over which you want to compute the correlation.

centeredoptions None | 0 | 1
default value = 1 removes the mean first. Set to 0 or None for uncentered.

biasedoptions None | 0 | 1
default value = 1 returns biased statistic. If want to compute unbiased statistic pass anything but 1.

max_pct_missingoptions
default value = 100. Maximum fraction of cell which is all masked.
Set to a percentage between 0 and 100%.
Set to 0. to mask results if any data is masked.
Set to 100. to calculate result if any data is not masked

covariance(x, y, weights=None, axis=0, centered=1, biased=1, max_pct_missing=100.0)

Function: covariance

Description of function:

Returns the covariance between 2 slabs. By default on the first dimension, centered and biased by default.

Usage:

```
cov = covariance(x, y, weights=weightoptions, axis=axisoptions,
                  centered=centeredoptions, biased=biasedoptions,
                  max_pct_missing=max_pct_missingoptions)
```

Options:

weightoptions
default = None. If you want to compute the weighted covariance, provide the weights here.

axisoptions 'x' | 'y' | 'z' | 't' | '(dimension_name)' | 0 | 1
default value = 0. You can pass the name of the dimension (integer value 0...n) over which you want to compute the covariance.

centeredoptions None | 0 | 1
 default value = 1 removes the mean first. Set to 0 or None
 uncentered.
 biasedoptions None | 0 | 1
 default value = 1 If want to compute an unbiased variance
 anything but 1.
 max_pct_missingoptions
 default value = 100. Maximum fraction of cell which is all
 Set to a percentage between 0 and 100%.
 Set to 0. to mask results if any data is masked.
 Set to 100. to calculate result if any data is not masked

geometricmean(x, axis=0, max_pct_missing=100.0)

Function: geometricmean

Description of function:

Returns the geometric mean over a specified axis.

Usage:

result = geometricmean(x, axis=axisoptions)

Options:

axisoptions 'x' | 'y' | 'z' | 't' | '(dimension_name)' | 0 | 1
 default value = 0. You can pass the name of the dimension
 (integer value 0...n) over which you want to compute the s
 max_pct_missingoptions
 default value = 100. Maximum fraction of cell which is all
 Set to a percentage between 0 and 100%.
 Set to 0. to mask results if any data is masked.
 Set to 100. to calculate result if any data is not masked

laggedcorrelation(x, y, lag=None, axis=0, centered=1, partial=1, biased=1, noloop=0, max_pct_missing=100.0)

Function: laggedcorrelation

Description of function:

Returns the correlation between 2 slabs at lag k centered, par
 "biased" by default.

Usage:

result = laggedcorrelation(x, y, lag=lagoptions, axis=axisopti
 centered=centeredoptions,
 partial=partialoptions,
 biased=biasedoptions, noloop=noloopoptions)

Returns value for x lags y by lag

Options:

lagoptions None | n | (n1, n2, n3...) | [n1, n2, n3]
 default value = None the maximum possible lags for specif
 is used.You can pass an integer, list of integers, or tupl
 axisoptions 'x' | 'y' | 'z' | 't' | '(dimension_name)' | 0 | 1
 default value = 0. You can pass the name of the dimension
 (integer value 0...n) over which you want to compute the s
 centeredoptions

```

        default value = 1 removes the mean first. Set to 0 or None
        uncentered
partialoptions None | 0 | 1
        default value = 1 uses only common time for means.
biasedoptions None | 0 | 1
        default value = 1 If want to compute an unbiased variance
        anything but 1.
noloopoptions None | 0 | 1
        default value = 0 computes statistic at all lags upto 'lag'
        set noloop=1 statistic is computed at lag only (not up to
max_pct_missingoptions
        default value = 100. Maximum fraction of cell which is all
        Set to a percentage between 0 and 100%.
        Set to 0. to mask results if any data is masked.
        Set to 100. to calculate result if any data is not masked

```

laggedcovariance(x, y, lag=None, axis=0, centered=1, partial=1, noloop=0, max_pct_missing=100.0)

Function: laggedcovariance

Description of function:

Returns the covariance between 2 slabs at lag k centered and p
default.

Usage:

```

result = laggedcovariance(x, y, lag=lagoptions, axis=axisoptions,
                           centered=centeredoptions,
                           partial=partialoptions, noloop=noloopoptions)

```

Returns value for x lags y by lag (integer)

Options:

```

lagoptions None | n | (n1, n2, n3...) | [n1, n2, n3 ....]
        default value = None the maximum possible lags for specif
        is used.You can pass an integer, list of integers, or tupl
        integers.
axisoptions 'x' | 'y' | 'z' | 't' | '(dimension_name)' | 0 | 1
        default value = 0. You can pass the name of the dimension
        (integer value 0...n) over which you want to compute the s
centeredoptions
        default value = 1 removes the mean first. Set to 0 or None
        uncentered
partialoptions None | 0 | 1
        default value = 1 uses only common time for means.
noloopoptions None | 0 | 1
        default value = 0 computes statistic at all lags upto 'lag'
        set noloop=1 statistic is computed at lag only (not up to
max_pct_missingoptions
        default value = 100. Maximum fraction of cell which is all
        Set to a percentage between 0 and 100%.
        Set to 0. to mask results if any data is masked.
        Set to 100. to calculate result if any data is not masked

```

linearregression(y, axis=None, x=None, error=None, probability=None, nointercept=None, noslope=None)

Computes the linear regression of y over x or an axis. This function returns values of the slope and intercept, and optionally, Error estimates associated probability distributions for T-value (T-Test) and F-value (analysis of variance f) can be returned. You can choose to return for either slope or intercept or both (default behaviour). For the details, refer to 'Statistical Methods in Atmospheric Sciences' by Daniel S. Wilks, Academic Press, 1995.

Usage:

```
result = linearregression(y, axis=axisoptions, x=xoptions,
```

Options:

```
axisoptions 'x' | 'y' | 'z' | 't' | '(dimension_name)' | 0 | 1
    default value = 0. You can pass the name of the dimension
    (integer value 0...n) over which you want to treat the array as
    dependent variable.
```

```
xvalues
```

```
    default = None. You can pass an array of values that are treated
    as the independent axis x
```

```
nointerceptoptions None | 0 | 1
```

```
    default = None. Setting to 0 or None means intercept calculations
    are returned. To turn OFF the intercept computations set
    nointercept = 1.
```

```
noslopeoptions None | 0 | 1
```

```
    default = None. Setting to None or 0 means slope calculations
    are returned. To turn OFF the slope computations set noslope to 1.
```

```
erroroptions None | 0 | 1 | 2 | 3
```

```
    default = None. If set to 0 or None, no associated errors are
    returned.
```

```
    If set to 1, the unadjusted standard error is returned.
```

```
    If set to 2, standard error returned. This standard error is
    adjusted using the centered autocorrelation of the residuals.
```

```
    If set to 3, standard error returned. The standard error is
    adjusted using the centered autocorrelation of the raw data.
```

```
probabilityoptions None | 0 | 1
```

```
    default = None. If set to 0 or None, no associated probabilities
    are returned. Set this to 1 to compute probabilities.
```

```
    Note: Probabilities are returned only if erroroptions are set to
    of 1, 2, or 3. If it is set to None or 0, then setting
    probabilityoptions has no meaning.
```

What is returned?

The returned values depend on the combination of options you specify. If both slope and intercept are required, a tuple is returned for slope and intercept and optionally Error (or optionally associated Probabilities), if erroroptions is set. If only one set (slope OR intercept) is required, values (not tuples) are returned. See examples below for more details.

When erroroption = 1 (from description above for erroroptions that means unadjusted standard error) and probabilityoptions = 1, the following are returned:

```
pt1 : The p-value for regression coefficient t-value. (With
```

```

        adjustment for standard error or critical t-value.)
None: There is only one p-value to be returned (pf1) but N
        returned to keep the length of the returned values c
pf1 : The p-value for regression coefficient F-value (one-
pf2 : The p-value for regression coefficient F-value (two-

```

When `erroroption = 2` or `3` (implying error adjustment using the `t`-value or the raw data and `probabilityoption = 1`, then the following values are returned:

```

pt1 : The p-value for regression coefficient t-value. (With
        sample size adjustment for standard error of slope.
pt2 : The p-value for regression coefficient t-value. (With
        sample size adjustment for standard error of slope a
        t-value.)
pf1 : The p-value for regression coefficient F-value (one-
pf2 : The p-value for regression coefficient F-value (two-

```

The values `pt1` and `pt2` are used to test the null hypothesis that y is independent of x (i.e., y is independent of x).

The values `pf1` and `pf2` are used to test the null hypothesis that the regression is linear (goodness of linear fit). For non-regression values of y , the degrees of freedom are 1 and $n-2$.

Examples:

Let us first examine the default behaviour of the `linearregression` function.

```
>>> Values = statistics.linearregression(y)
```

#The returned "Values" is actually a tuple consisting of the slope and intercept. They can also be accessed as follows:

```
>>> slope, intercept = statistics.linearregression(y)
```

If error estimates are also required, then:

```
>>> Values, Errors = linearregression(y, error=1)
```

#where "Values" and "Errors" are tuples containing answer for slope AND intercept. You can break them as follows.

```

#slope, intercept = Value and slope_error, intercept_error = E
>>> (slope, intercept), (slo_error, int_error) =

```

#WARNING: The following will not work.

```
>>> slope, intercept, slo_error, int_error = linearregression(y,
```

#To get the standard error non adjusted result for slope only

```
>>> slope, slope_error = linearregression(y, error=1, nointerc
```

#In the line below all the returned values are tuples.

```
>>> Values, Errors, Pt1, Pt2, Pf1, Pf2 = linearregression(y,
```

#That means in the above statement is returning tuples ordered as follows:
#(slope, intercept), (slo_error, int_error), (pt1_slo, pt1_int)

```
#If we want results returned for the intercept only.  
>>> intercept,intercept_error,pt1,pt2,pf1,pf2 =
```

meanabsdiff(x, y, weights=None, axis=0, centered=1, max_pct_missing=100.0)

Function: meanabsdiff

Description of function:

Returns the mean absolute difference between 2 slabs x and y.
on the first dimension and centered

Usage:

```
result = meanabsdiff(x, y, weights=weightoptions, axis = axisoptions,  
                      centered=centeredoptions)
```

Options:

weightoptions

default = None returns equally weighted statistic. If you
compute the weighted statistic, provide weights here.

axisoptions 'x' | 'y' | 'z' | 't' | '(dimension_name)' | 0 | 1

default value = 0. You can pass the name of the dimension
(integer value 0...n) over which you want to compute the s

centeredoptions None | 0 | 1

default value = 1 removes the mean first. Set to 0 or None

max_pct_missingoptions

default value = 100. Maximum fraction of cell which is all
Set to a percentage between 0 and 100%.

Set to 0. to mask results if any data is masked.

Set to 100. to calculate result if any data is not masked

median(x, axis=0)

Function: median

Description of function:

Returns the median value of an array.

Usage:

```
result = median(x, axis=axisoptions)
```

Options:

axisoptions 'x' | 'y' | 'z' | 't' | '(dimension_name)' | 0 | 1

default value = 0. You can pass the name of the dimension
(integer value 0...n) over which you want to compute the s

percentiles(x, percentiles=[50.0], axis=0)

Function: percentiles

Description of function:

Returns values at the defined percentiles for an array.

Usage:

```
result = percentiles(x, percentiles=percentileoptions, axis=a
```

Options:

percentileoptions A python list of values
 Default = [50.] (the 50th percentile i.e the median value)
 axisoptions 'x' | 'y' | 'z' | 't' | '(dimension_name)' | 0 | 1
 default value = 0. You can pass the name of the dimension
 (integer value 0...n) over which you want to compute the statistic

rank(x, axis=0)

Function: rank

Description of function:

Returns the rank of each element along the specified axis
 where 0 is lowest value, 100 is maximum value. Handles missing values

Usage:

result = rank(x, axis=axisoptions)

Options:

axisoptions 'x' | 'y' | 'z' | 't' | '(dimension_name)' | 0 | 1
 default value = 0. You can pass the name of the dimension
 (integer value 0...n) over which you want to compute the statistic

rms(x, y, weights=None, axis=0, centered=0, biased=1, max_pct_missing=100.0)

Function: rms

Description of function:

Returns the root mean square difference between 2 slabs. By default
 a slab (on first dimension) "uncentered" and "biased" by default

Usage:

result = rms(x, y, weights=weightoptions, axis = axisoptions,
 centered=centeredoptions, biased = biasedoptions,
 max_pct_missing=max_pct_missingoptions)

Options:

weightoptions
 default = None returns equally weighted statistic. If you
 compute the weighted statistic, provide weights here.
 axisoptions 'x' | 'y' | 'z' | 't' | '(dimension_name)' | 0 | 1
 default value = 0. You can pass the name of the dimension
 (integer value 0...n) over which you want to compute the statistic
 centeredoptions None | 0 | 1
 default value = 0 returns uncentered statistic (same as No
 remove the mean first (i.e centered statistic) set to 1. No
 other statistic functions return a centered statistic by default
 biasedoptions None | 0 | 1
 default value = 1 If want to compute an unbiased variance
 anything but 1.
 max_pct_missingoptions
 default value = 100. Maximum fraction of cell which is all
 Set to a percentage between 0 and 100%.
 Set to 0. to mask results if any data is masked.
 Set to 100. to calculate result if any data is not masked

std(x, weights=None, axis=0, centered=1, biased=1, max_pct_missing=100.0)

Function: std

Description of function:

Returns the standard deviation from a slab. By default on first dimension, centered, and biased.

Usage:

```
result = std(x, weights=weightoptions, axis = axisoptions,
             centered=centeredoptions, biased = biasedoptions,
             max_pct_missing=max_pct_missingoptions)
```

Options:

weightoptions

If you want to compute the weighted statistic, provide weights. axisoptions 'x' | 'y' | 'z' | 't' | '(dimension_name)' | 0 | 1 default value = 0. You can pass the name of the dimension (integer value 0...n) over which you want to compute the statistic.

centeredoptions None | 0 | 1

default value = 1 removes the mean first. Set to 0 or None uncentered.

biasedoptions None | 0 | 1

default value = 1 If want to compute an unbiased variance anything but 1.

max_pct_missingoptions

default value = 100. Maximum fraction of cell which is all masked. Set to a percentage between 0 and 100%.

Set to 0. to mask results if any data is masked.

Set to 100. to calculate result if any data is not masked

variance(x, weights=None, axis=0, centered=1, biased=1, max_pct_missing=100.0)

Function: variance

Description of function:

Returns the variance from a slab. By default on first dimension, centered, and biased.

Usage:

```
result = variance(x, weights=weightoptions, axis = axisoptions,
                  centered=centeredoptions, biased = biasedoptions,
                  max_pct_missing=max_pct_missingoptions)
```

Options:

weightoptions

If you want to compute the weighted variance, provide weights. axisoptions 'x' | 'y' | 'z' | 't' | '(dimension_name)' | 0 | 1 default value = 0. You can pass the name of the dimension (integer value 0...n) over which you want to compute the statistic.

centeredoptions None | 0 | 1

default value = 1 removes the mean first. Set to 0 or None uncentered.

biasedoptions None | 0 | 1

default value = 1 If want to compute an unbiased variance anything but 1.

max_pct_missingoptions

default value = 100. Maximum fraction of cell which is all masked. Set to a percentage between 0 and 100%.



Set to 0. to mask results if any data is masked.
Set to 100. to calculate result if any data is not masked